IP Packet

<table>
<tr><th>0 3</th><th>4 7</th><th>8 15</th><th colspan="3">16</th><th>31</th></tr>
<tr><td>version
ip_v</td><td>hdr. len.
ip_hl</td><td>type of service
ip_tos</td><td colspan="4">total length (bytes)
ip_len</td></tr>
<tr><td colspan="3">identification
ip_id</td><td>0</td><td>D
F</td><td>M
F</td><td>fragment offset
ip_off</td></tr>
<tr><td colspan="2">time to live
ip_ttl</td><td>protocol
ip_p</td><td colspan="4">IP header checksum
ip_sum</td></tr>
<tr><td colspan="7">source address
ip_src</td></tr>
<tr><td colspan="7">destination address
ip_dst</td></tr>
<tr><td colspan="7">IP options (if any)</td></tr>
<tr><td colspan="7">data</td></tr>
</table>

Version

4 IP "Version 4"

Header Length

In 32 bit words. Multiply by 4 for length in bytes.

Type of Service (PreDTRC)

Bits 8–10:	Precedence
Bit 11:	minimum delay
Bit 12:	maximize throughput
Bit 13:	maximize reliability
Bit 14:	minimize cost
Bit 15:	reserved

Type of Service (Differentiated Services)

Bits 8–13:	0
Bit 14:	ECN capable
Bit 15:	congestion notification

Flags

DF	Don't Fragment
MF	More Fragments

Protocol

1	ICMP	9	IGRP	51	AH
2	IGMP	17	UDP	88	EIGRP
4	IP/IP	47	GRE	89	OSPF
6	TCP	50	ESP	115	L2TP

TCP Packet

<table>
<tr><th colspan="8">0 15</th><th>16 31</th></tr>
<tr><td colspan="8">source port
th_sport</td><td>destination port
th_dport</td></tr>
<tr><td colspan="9">sequence number
th_seq</td></tr>
<tr><td colspan="9">acknowledgment number
th_ack</td></tr>
<tr><td>offset
th_off</td><td>reserved</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td>window size
th_win</td></tr>
<tr><td colspan="8">checksum
th_sum</td><td>urgent pointer
th_urp</td></tr>
<tr><td colspan="9">options (if any)</td></tr>
<tr><td colspan="9">data (if any)</td></tr>
</table>

Offset

Four bit header length in 32 bit words. It's the offset to where the data begins. Multiply by 4 for length in bytes.

Reserved

Must be zero unless ECN is used, in which case:

Bits 4–7:	0
Bit 8:	CWR (TH_CWR)
Bit 9:	ECN-Echo (TH_ECE)

Flags

URG	TH_URG
ACK	TH_ACK
PSH	TH_PUSH
RST	TH_RST
SYN	TH_SYN
FIN	TH_FIN

Options

0	end of options	3	window scale
1	pad (no operation)	4	SACKs OK
2	MSS	8	time stamp

States

0	CLOSED	6	FIN-WAIT-1
1	LISTEN	7	CLOSING
2	SYN-SENT	8	LAST-ACK
3	SYN-RCVD	9	FIN-WAIT-2
4	ESTABLISHED	10	TIME-WAIT
5	CLOSE-WAIT		

UDP Packet

<table>
<tr><th>0 15</th><th>16 31</th></tr>
<tr><td>source port
uh_sport</td><td>destination port
uh_dport</td></tr>
<tr><td>length
uh_ulen</td><td>checksum
uh_sum</td></tr>
<tr><td colspan="2">data (if any)</td></tr>
</table>

sockaddr_in Structure

<table>
<tr><th>0 7</th><th>8 15</th><th>16 31</th></tr>
<tr><td>len
sin_len</td><td>family
sin_family</td><td>port
sin_port</td></tr>
<tr><td colspan="3">in_addr structure
sin_addr</td></tr>
<tr><td colspan="3">Reserved (8 bytes)
sin_zero</td></tr>
</table>

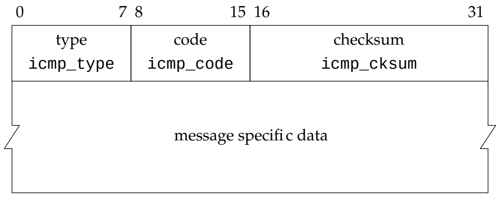
Family

AF_INET	Internet
AF_UNIX	IPC

in_addr Structure

<table>
<tr><th>0 31</th></tr>
<tr><td>IP address
s_addr</td></tr>
</table>

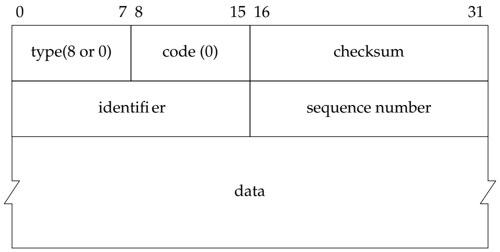
ICMP Packet



Types and Codes

0		Echo reply
3		Destination unreachable
	0	Net unreachable
	1	Host unreachable
	2	Protocol unreachable
	3	Port unreachable
	4	Fragmentation needed, DF set
	5	Source route failed
	6	Destination network unknown
	7	Destination host unknown
	8	Source host isolated
	9	Network admin. prohibited
	10	Host admin. prohibited
	11	Network unreachable for TOS
	12	Host unreachable for TOS
	13	Comm. admin. prohibited
	14	Host precedence violation
	15	Precedence cutoff in effect
4		Source quench
5		Redirect
	0	Network
	1	Host
	2	TOS and network
	3	TOS and host
8		Echo request
9		Router advertisement
10		Router selection
11		Time exceeded
	0	TTL exceeded in transit
	1	Fragment reassembly time
12		Parameter problem
	0	Pointer indicates error
	1	Missing required option
	2	Bad length
13		Timestamp
14		Timestamp reply
15		Information request
16		Information reply
17		Address mask request
18		Address mask reply
30		Traceroute

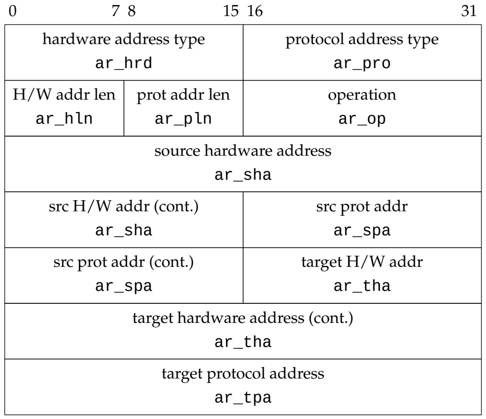
Ping Packet



Identifier
The pid of the process issuing the `ping`

Sequence Number
Number used to tie the request to reply. Incremented by 1 for each request.

ARP Packet



Hardware address type

1	Ethernet (ARPHRD_ETHER)
6	Token-ring (ARPHRD_IEEE802)
15	frame relay (ARPHRD_FRELAY)

Hardware address length
6 for Ethernet (shown)

Protocol address length
4 for IPv4 (shown)

Operation

1	resolve addr request (ARPOP_REQUEST)
2	resolve addr reply (ARPOP_REPLY)
3	prot addr request (ARPOP_REVREQUEST)
4	prot addr reply (ARPOP_REVREPLY)
8	ident peer request (ARPOP_INVREQUEST)
9	ident peer reply (ARPOP_INVREPLY)

TCP/IP
Pocket Guide

```
int accept( int s, struct sockaddr *addr,
   int *addrlen )
int bind( int s, const struct sockaddr *name,
   int namelen )
int connect( int s, const struct sockaddr *peer,
   int peerlen )
int listen( int s, int backlog )
int recv( int s, void *buf, size_t len, int flags )
int recvfrom( int s, void *buf, size_t len, int flags,
   struct sockaddr *from, int *fromlen )
int send( int s, const void *buf, size_t len,
   int flags )
int sendto( int s, const void *buf, size_t len,
   int flags,
   const struct sockaddr *to, int tolen )
int shutdown( int s, int how )
int socket( int domain, int type, int protocol )
```

flags

CMSG_OOB	1
CMSG_PEEK	2
CMSG_DONTROUTE	4

domain

AF_INET	Internet
AF_LOCAL	IPC
AF_UNIX	IPC (same as AF_LOCAL)

type

SOCK_STREAM	stream socket
SOCK_DGRAM	datagram socket
SOCK_RAW	raw socket

Revised 28 Aug 02